Rich Context Competition using Attention based BiLSTM and CRFs

Ben Zhang
New York University
Deparmtnet of Data Science
bz957@nyu.edu

Ji Chen
New York University
Department of Population Health
ji.chen@nyulangone.org

Angelina Volkova
New York University
School of Medicine
angelina.volkova@nyulangone.org

Abstract

We used attention-based mechanism BiLSTM CRF to build a Named Entity Recognition Model which can automate the discovery of mentioned research data sets in document level, then according the mention list to find the cited dataset in social science publications. After that we applied Bag of Words and LDA to infer the research methods and associated fields of study separately.

1 Introduction

As numerous research studies are being published actively in various research fields using self-generated or publicly available data sets, it is not easy for researchers and analysts who aim to use specific data of interest to find out published work with similar sources of data used. Often, some valuable research studies are lost in the gigantic pool of publications and redundant empirical work is being carried out without knowing what has been established. In order to tackle this difficulty, the ongoing "Rich Context Competition" was held by New York University's Coleridge Initiative. Participants are expected to develop text analysis and machine learning techniques to explore relationships between data sets, researchers, publications, research methods, and fields. The task is to automate the discovery of mentioned research data sets, and inference of research methods and associated fields of study in social science publications. The results from the participants will be used to create a rich context for empirical research, and build novel metrics to describe data use. In this report, we discuss our approaches and the corresponding results.

2 Methods

2.1 Data

Participants are provided with 5,000 training publications and 100 validation publications in both pdf and text format along with the corresponding metadata in json format. A test set of 5,000 corpus will be used to evaluate submitted models. The text format corpus was generated using the open source Xpdf text extraction system using the following command:

```
pdftotext -raw <path_to_pdf.pdf>
<path_to_txt.txt>
```

Target labels for data set mentions are provided as exact mentioned phrases (from now on referred to as "mention lists") and data set IDs in json format. In addition, a global data set vocabulary of all mention lists. A method vocabulary and a field vocabulary are provided as reference for unsupervised inference in json format but novel methods and fields are also encouraged.

2.2 Preprocessing

2.2.1 Data set identification

Raw texts were parsed into sentences using the NLTK toolkit, and further tokenized into word tokens per sentence. Punctuations in both training corpus and labeled mention lists were removed. Tokens containing non-printable characters were removed. Max sentence length was set to 30 covering most sentences while balancing with amount of padding downstream (Figure 1). Sentences longer than 30

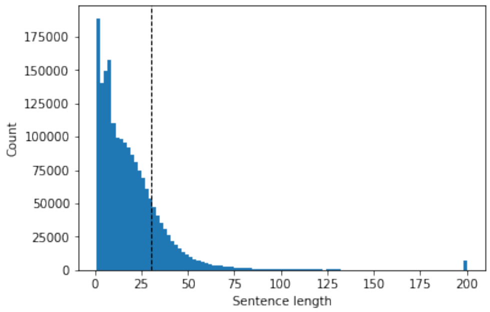


Figure 1: Histogram of clipped (200) sentence length from all sentences in the 5,000 training corpus.

tokens were cut into multiple sentences of 30 tokens. Extremely long sentences were mostly caused by context from tables. However, due to cases where mention lists are located in tables, we decided not to discard tables for the purpose of generalization and accurate performance measure.

For each word token, a tag was generated to represent the word besides embedding representation which was included during model training. Four different tags "B, I, O, M" were used, where "B" indicates "beginning" of an entity, "I" indicates "In (within)" an entity, "O" indicates "Outside" of an entity and "M" indicates informative tokens in the "Mentioned" sentences where an entity is mentioned. Informative tokens were picked from the top 100 frequent tokens of all words in mentioned sentences via Bag-Of-Word count based quantification after removing stop words.

2.2.2 Method and field inference

Different from the data set identification task, the rest two tasks do not require explicit location information of word tokens but are rather similar to text classification (field) and word synonym identification. We decided to tokenize all words with lemmatization and remove all punctuations and stop words. Additionally, tokens not in the English language or shorter than 4 characters were removed. For field inference, only the first 10,000 characters were extracted to train our model.

2.3 Experiment design

2.3.1 Data set identification

This task was broken down into two parts: 1) NER: text corpus was parsed into token level and each token was tagged with NER labels as well as informative representations listed in 2.2.1. 2) Predicted tokens with NER tags in each sentence were concatenated as predicted "mention lists" which were later compared with global mention lists respectively using cosine similarity. The data set with the best matching mention list will be the final prediction. The 5,000 training corpus was split into 4,500 train set and 500 validation set. The 100 corpus was used as a hold-out set for performance report.

2.3.2 Method and field inference

Method inference and field inference were approached as unsupervised learning, with additional reference from each vocabulary. All 5,000 corpus was used for building the models in each task respectively.

2.4 Algorithms

2.4.1 Data set identification

We used a hybrid of deep neural networks (LSTM/GRU/CNN) with conditional random fields (CRFs), a sequence model which used for structured prediction, is widely used in the field of NER. In addition, we explored a couple of embedding models and tagging representations as follows:

- Word Embedding

Compared with word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), FastText breaks words into the character level, which allows rare words be represented as a summary of character embeddings. By this method, some acronyms can be captured by part of their n-grams.We trained 100 dimensional word embeddings with the whole 5,000 training corpus using the skip-gram model in the Gensim FastText package.

- Character-level embedding

Character-level features can not only alleviate out-of-vocabulary problem mainly caused by prefixes, suffixes or acronyms, but also enable the model to learn the structure of a word. We keep the upper case characters in each word to pass this information to our model because if a word or consecutive words in the middle of a sentence contain characters upper case they are likely entities of interest, e.g

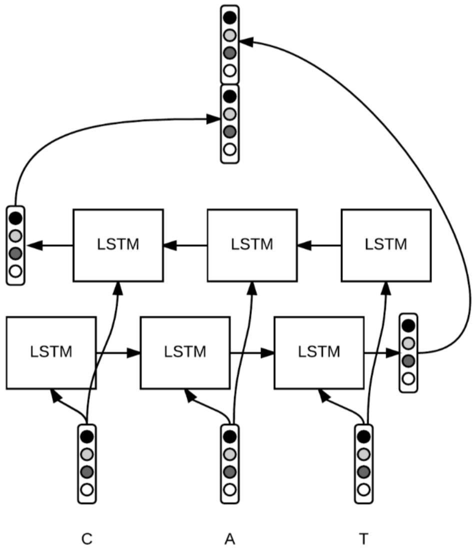


Figure 2: The bidirectional Long Short-Term Memory (BiLSTM) architecture for character embedding.

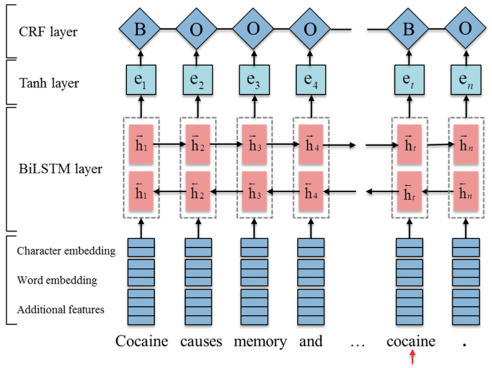


Figure 3: BiLSTM-CRF model structure without attention mechanism.

"NHDS" or "National Hospital Discharge Survey". We initialized a random character lookup table and loop it into a bidirectional Long Short-Term Memory (BiLSTM) architecture (Figure 2).

- Vanila BiLSTM/GRU-CRF model

Among the variants of CRF-based approaches the bidirectional LSTM and CRFs is one of the most popular models (Figure 3).

We concatenated our pre-trained FastText word embedding, character embedding from BiLSTM. Each sentence was represented as a sequence of vectors X=(x1,. . . ,xt,. . . ,xn) which was fed into a BiRNN(LSTM/GRU) layer and concatenated into the hidden vector hi [hi,hi]. The output of each hidden layer hi was et=tanh(Weht) In the CRF layer, the output of each word representation after BiLSTM was a Matrix P, where Pi, yi represent the probability from xi to tag yi. Another Matrix Ayi Ay+1 the score of transition from tag i to tag j in consecutive words. s(X,y) is the total predicted score of yi. We used softmax to transform the scores and used the negative-log likelihood to as the loss function.

$$s(\mathbf{X},\mathbf{y}) = \sum_{i=0}^{n} A_{y_i,y_{i+1}} + \sum_{i=1}^{n} P_{i,y_i}$$

Equation 1.

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X},\mathbf{y})}}{\sum_{\widetilde{\mathbf{y}}\in\mathbf{Y_X}} e^{s(\mathbf{X},\widetilde{\mathbf{y}})}}.$$

Equation 2.

$$\begin{aligned} log(p(y|X)) &= log(\frac{e^{S(X,y)}}{\sum_{\widetilde{y}\in Y_X} e^{S(X,\widetilde{y})}}) \\ &= S(X,y) - log(\sum_{\widetilde{y}\in Y_X} e^{S(X,\widetilde{y})}) \end{aligned}$$

Equation 3.

- Attention based BiLSTM model

We added an attention layer between BiLSTM CRF to allow the model capture similar tokens from previous tokens at the document level.

For an input document D=(X1,. . . ,Xt,. . . ,Xm) consisting of m sentences, each sentence was expressed as X=(x1,. . . ,xt,. . . ,xn). The total length of the Document is N=m*n. We pre-computed the similarity score xi with xj N at each time step t by cosine similarity. Wa is a differentiable parameter.

$$\text{score}(\mathbf{x}_t,\mathbf{x}_j) = \frac{\mathbf{W}_a(\mathbf{x}_t \bullet \mathbf{x}_j)}{|\mathbf{x}_t||\mathbf{x}_j|}$$

Equation 4.

The attention weight value t, j between word xi with xj N was calculated by softmax.

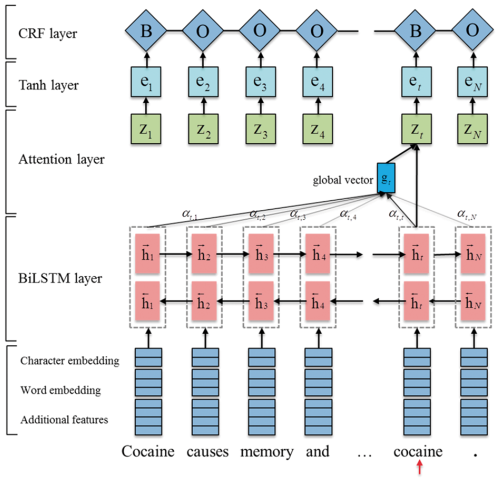


Figure 4: BiLSTM-CRF model structure without attention mechanism.

$$\alpha_{t,j} = \frac{\exp\ (score(\mathbf{x}_t,\mathbf{x}_j))}{\sum_k \exp\ (score(\mathbf{x}_t,\mathbf{x}_k))}$$

Equation 5.

A global vector for word xi at time step t was computed by summing up the product of BiLSTM output hj and the attention weight t,j:

$$\mathbf{g}_t = \sum_{j=1}^{N} \alpha_{t,j}\mathbf{h}_j$$

Equation 6.

After that we use tanh function on the concatenated vector gt and ht and get zt. The following step of CRF layer is same as mentioned above.

$$\mathbf{z}_t = \tanh(\mathbf{W}_g[\mathbf{g}_t;\mathbf{h}_t])$$

Equation 7.

2.4.2 Method inference

A search engine based approach was used to identify methods mentioned in the corpus. Up-to 3-gram tokens were used to match the method vocabulary.

2.4.3 Field inference

We used a generative statistical model, latent Dirichlet allocation (LDA) to infer fields of study.

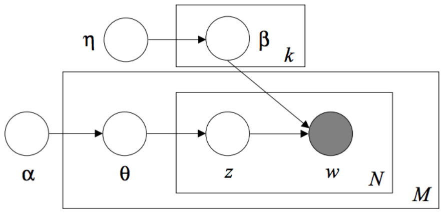


Figure 5: Graphical model representation of the smoothed LDA model.

In general, the LDA attempts to explain observations by predefined, unobserved groups and outputs signature tokens representative of each group. The dependencies among parameters trained in the LDA model can be depicted in figure x (Blei et al. 2003). M denotes the corpus universe and N denotes the token universe in each corpus.   indicates the topic distribution per corpus, indicates the token distribution per topic group, indicates the distribution of the k-th topic group, indicates the topic distribution for corpus m where m M, z indicates the topic group for word n where n N, and w indicates a specific token in N. The training process was initialized by randomly assigning each word token in each corpus to one of the 10 fields in the given vocabulary. For each n, the proportion of w in n that are currently assigned to (p(—n)) and the proportion of w count per in M (p(w—)) were computed. Each w was reassigned to according to the product of p(—n) and p(w—) and the two proportions were re-computed each step until the assignments stabilize. The model outputs a predefined number of signature tokens per topic group. In this task, the cosine similarity of the output and the reference field definition was computed using pre-trained general English word embedding from Spacy. The top match was picked as the inferred field.

2.5 Evaluation metrics

Evaluation metrics are based on standard binary classification precision, recall and F1 measure. Recall is defined as Recall = (of true positives) / ((of true positives) + (of false negatives)). Precision is defined as Precision = (of true positives) / ((of true positives) + (of false positives)) and F1 is defined as F1 = 2 / ((1 / Recall) + (1 / Precision)). For our own purpose, we also measure precision, recall, and F1 measure for our NER tag prediction.

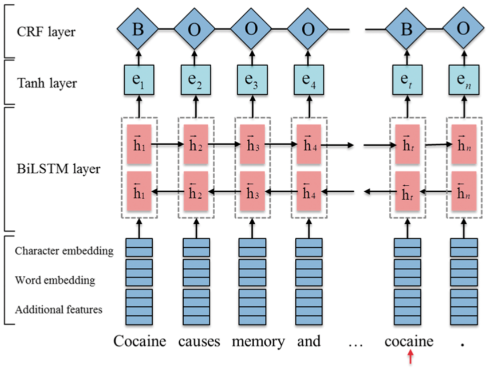


Figure 6: BiLSTM-CRF model structure without attention mechanism.

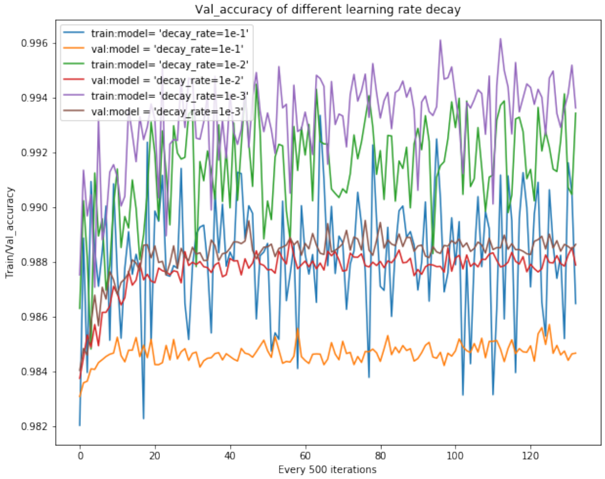


Figure 7: Accuracy of train set and validation set using different learning rate decay. "Train lr = 1e-1": training accuracy with learning rate decay 0.1; .

3 Results

3.1 Data set identification

3.1.1 Parameter tuning

Since the total number of words in training data set before padding is 38 million,due to the computational capacity and training time limitation, we regretfully to random sample 1/10 size data to tune our six parameters in our vanila BiLSTM/GRU-CRF models: learning rate, dropout rate, activation function, optimizer, rnn model type.

The decay of learning rate affects the accuracy more than other parameters. Among them the decay rate=0.01 acchieve best result.

Other parameters such as activation, dropout rate make some impact on the final result. But not as

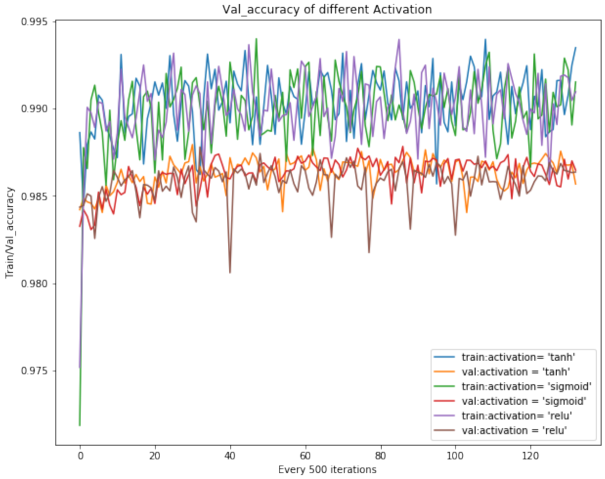


Figure 8: Accuracy of train set and validation set using different activation. "Train activation = tanh": training accuracy with activation function 'tanh' in dense layer .

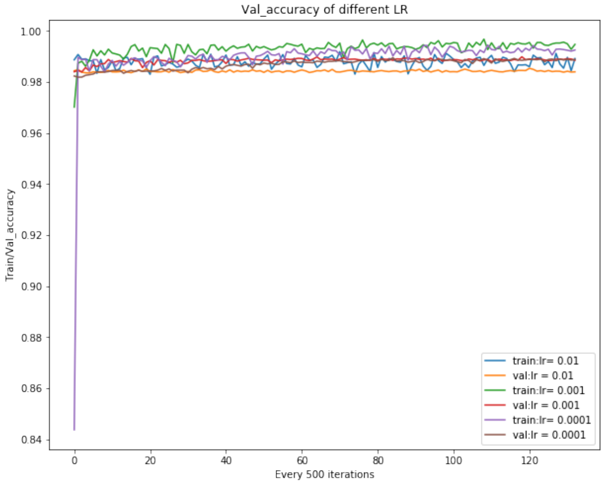


Figure 9: Accuracy of train set and validation set using different learning rate. "Train lr = 0.01": training accuracy with learning rate 0.01; .

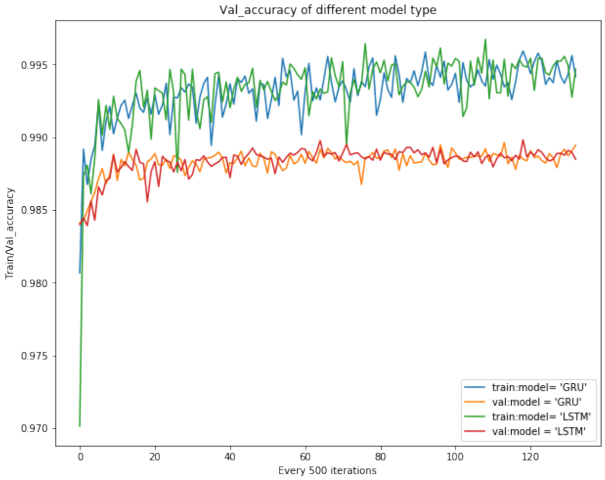


Figure 10: Accuracy of train set and validation set using different RNN model. “Train model = ’GRU’”: training accuracy with ’GRU’ RNN model.

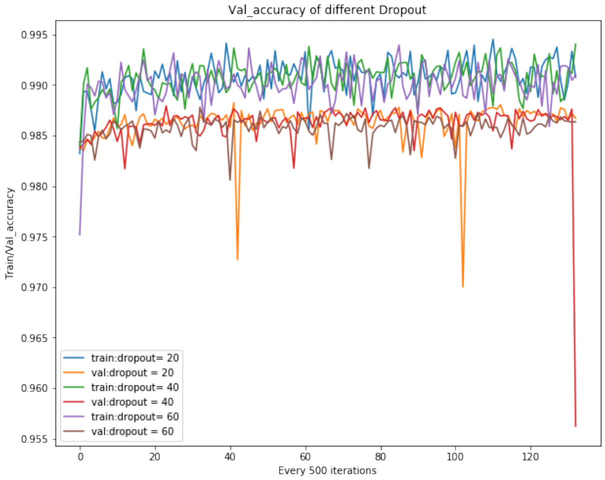


Figure 11: Accuracy of train set and validation set using different dropout rate. “Train dropout = 20”: training accuracy with dropout rate 20%; .

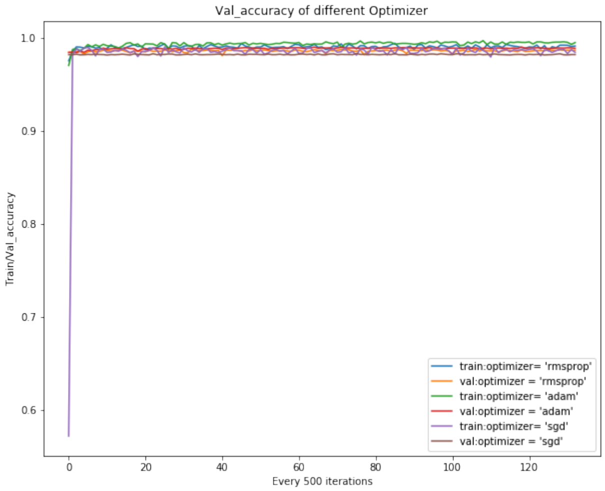
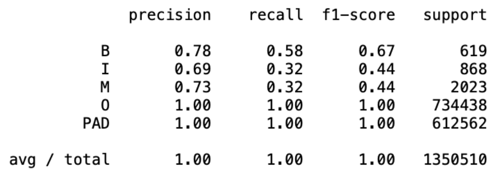


Figure 12: Accuracy of train set and validation set using different optimizer. “Train optimizer= rmsprop”: training accuracy with ’rmsprop optimizer’; .



```
             precision    recall  f1-score   support

          B       0.78      0.58      0.67       619
          I       0.69      0.32      0.44       868
          M       0.73      0.32      0.44      2023
          O       1.00      1.00      1.00    734438
        PAD       1.00      1.00      1.00    612562

avg / total       1.00      1.00      1.00   1350510
```

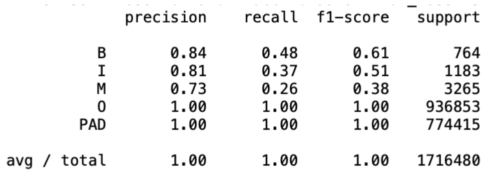
Figure 13: NER result of tuned model without attention mechanism.

impactful as learning rate decay.

The attention mechanism let the model get higher precision on all target tags and recall in tag ’I’ which has bigger amount in our dataset. It shows the attention mechanism helps but not a lot considering the huge computational cost it brings.

3.2 Method inference

By using a baseline search method, we found overall 75,245 method matches in the 5,000 training corpus and in the 100 hold out set. With a better design of the method, such as training an embedding of n-gram word vectors, more methods could be inferred



```
             precision    recall  f1-score   support

          B       0.84      0.48      0.61       764
          I       0.81      0.37      0.51      1183
          M       0.73      0.26      0.38      3265
          O       1.00      1.00      1.00    936853
        PAD       1.00      1.00      1.00    774415

avg / total       1.00      1.00      1.00   1716480
```

Figure 14: NER result of tuned model with attention mechanism.

by a thorough search of highly similar n-grams with the method vocabulary. In addition, by introducing novel methods into the vocabulary, we will be able to infer a larger coverage of all mentioned methods in the global corpus.

3.3 Research field inference

The LDA model was trained on the first 10,000 characters with the assumption that the word tokens extracted contain sufficient information for this task. The mean probability score of all of the publications assignments was 0.69, with the highest score larger than 0.95. The assignments of publications whose main language of the corpus were not English had poor results. Example signature word token sets are listed as below:
1) bank media study market article paper economic news bundesbank author
2) women sexual study evolutionary psychology partner university mate mat parent
3) health study survey national data risk state report adults population
Overall the model performs well. The above three example token sets can be assigned to “economy”, “psychology”, and ”hospitalization” respectively by eyeballing. However, publications are likely not limited to the 10 general research fields in the vocabulary. Further improvements of the model could include adding novel fields into the vocabulary, using external data and incorporating more tokens per publication with allowed computational resources.

4 Discussion

Our data set identification model is able to identify basic patterns of NER tags but fails generalize to the hold out set. One possible reason is that the tags are extremely imbalanced with a 0.1 percent of ”I” tag. Moreover, the attention mechanism we added did not give us a performance boost. By calculating the cosine similarity globally, it was computationally inefficient too. A better design of the attention mechanism will surely help improve the model. Additionally, we have not explored our options such as: 1) Adding Part-Of-Speech tags onto each word token; 2) Adding external data; 3) Potentially better experiment design.

5 Team member contribution

Ben Zhang: Preprocessing, data set identification, method inference
Ji Chen: Preprocessing, method inference, field inference, algorithm discussion in data set identification, method inference and field inference
Angelina Volkova: Field inference

6 Github Repository

`https://github.com/jichen1010/NLP_2018_project.git`

Acknowledgments

We thank Dr. Cho, NYU HPC Prince and the Rich Context Competition committee for supporting our research work

References

1,Blei, David M.; Ng, Andrew Y.; Jordan, Michael I (January 2003). Lafferty, John, ed. ”Latent Dirichlet Allocation”. Journal of Machine Learning Research. 3 (4–5): pp. 993–1022. doi:10.1162/jmlr.2003.3.4-5.993

2,Ling Luo;Zhihao Yang; Pei Yang; Yin Zhang; Lei Wang Hongfei;Lin Jian Wang.”An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition”. Bioinformatics, Volume 34, Issue 8, 15 April 2018, Pages 1381–1388.